ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND

HANSCOM AIR FORCE BASE, MASSACHUSETTS

MCI-77-2

March 1976

MULTILEVEL SECURITY FOR THE E-4 BLOCK II ADP

Approved for public release; distribution unlimited

DIRECTORATE OF COMPUTER SYSTEMS ENGINEERING

DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS

LEGAL NOTICE

When U.S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use or sell any patended invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

ANDREW H. FRENCH, Captain, USAF
Project Officer
Techniques Engineering Division

ROGER R. SCHELL, Lt Col, USAF
ANF System Security Program Manager

FOR THE COMMANDER

FRANK J. EMMA, Colonel, USAF
Director, Computer Systems Engineering
Deputy for Command & Management Systems

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. None	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MULTILEVEL SECURITY FOR THE E-4 BLOCK II ADP		5. TYPE OF REPORT & PERIOD COVERED Technical Paper
		6. PERFORMING ORG. REPORT NUMBER MCI-77-2
7. AUTHOR(s) Paul A. Karger, 1Lt, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Deputy for Command & Management Systems (MCI) Electronic Systems Division (AFSC) Hanscom AFB, Bedford, Massachusetts 01731		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE 64740F, Project 2239
11. CONTROLLING OFFICE NAME AND ADDRESS See Item 9		12. REPORT DATE March 1976
		13. NUMBER OF PAGES 69
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
This document is NOT AVAILABLE THROUGH DDC.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Secure Computer Systems
E-4
Multilevel Systems
ADP
Manually Released Transmission

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This document examines the Manually Released Transmission (MRT) approach to security in the E-4 Block II ADP and identifies implementation alternatives for achieving security with minimum cost and risk. It was prepared originally in response to a request from the E-4 Program Office, ESD/YS. This document is being published to serve as an example and guide in future acquisitions involving secure computer systems.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

1. INTRODUCTION

This paper will examine the Manually Released Transmission (MRT) approach to security in the E-4 Block II ADP and identify implementation alternatives for achieving security with minimum cost and risk. Since security must be an integral part of the E-4 Block II ADP, absolute cost estimates are not possible. However cost estimates relative to the various options for Block II can be provided. Information on the technical and cost aspects was obtained from the E-4 System Development Plan (SDP) [5] and from E-4 SPO personnel. This paper outlines representative options for providing security using the MRT approach. As the design for Block II becomes better defined, other options differing in detail from these may become more appropriate. Options for both Generic Levels A and D will be outlined with relative technical risk assessments included. The paper pays particular attention to minimizing risk for the Block II ADP by providing "fail soft" options in with recovery modes in case prerequisite technology results are delayed or otherwise not available to the E-4 probram. The paper also investigates security failure modes to attempt to insure that a single human mistake (an inadvertant mistake as opposed to a deliberate malicious attempt) cannot lead to compromise of information.

Three major classes of solutions are addressed for the Block II ADP security problem in the contexts of both generic Levels A and D:

a) a stand-alone MRT system separate from the CPE and ADP equipments (Para 2),

b) an MRT system integrated with the CPE (Generic Level A) (Para 3),

c) an MRT system integrated with th a CPE-ADP combination (Generic Level D) (Para 4).

Each of these major classes is further broken into several levels of cost and risk with capabilities for growth in security and WWMCCS compatibility outlined.

2. STAND-ALONE MRT SYSTEMS

2.1 Overview

The basic concept of a stand-alone Manual Released Transmission (MRT) system is described in [5]. It is based on an assumption that the hardware/software in the CPE and

(optional) ADP segment do not provide effective security controls and cannot be trusted to correctly mark and maintain the security levels of messages. To achieve the required security, the MRT system must ensure that every message sent out from the E-4 is reviewed manually by a human being who is knowledgable of the content and format of the messages and can ensure that the message is correctly classified. In a stand-alone MRT system, the human being is the only security control.

2.2 Alternative Implementations

There are three basic classes of stand-alone MRT implementations which will be described here. Variations are possible on the approaches shown and may be required as the Block II ADP becomes better defined. Each implementation has different construction costs, risks, and costs to certify for security.

2.2.1 Hardware/Firmware Implementation

This approach, which is described in the E-4 SDP, uses microprogrammed processors in the line control modules (LCMs), user display stations (UDSs), and secure interlocked switches (SIS's) to achieve security. (See Figure 1). Messages coming up to the E-4 are received by the LCM which buffers the message and passes it to the CPE. The LCM must ensure that no data received from the CPE is transmitted to the ground. This restriction includes "control" signals from the CPE, because malicious software in the CPE cooperating with the ground could communicate classified information by "modulating" control signals. For example, a message ACK protocol could be surreptitiously modified to transmit information as follows: If the message is ACK'ed once, then the CPE is transmitting a zero. If the message is ACK'ed twice, then the CPE is transmitting a one. Using such a protocol, malicious software introduced into the CPE in a clandestine modification could transmit several bits per second without detection. The solution to such a problem is not identified in the SDP which states a requirement, however that, "There shall be no capability for transfer of data (other than control signals) from the data processing functions to the link controller". (1) This does not necessarily mean the concept is invalid, since it is theoretically possible to build an LCM which ensures that even control signals from the CPE do not reach the ground. The question of practicality, however, raises the risk of

---

(1) [5], page 3-24

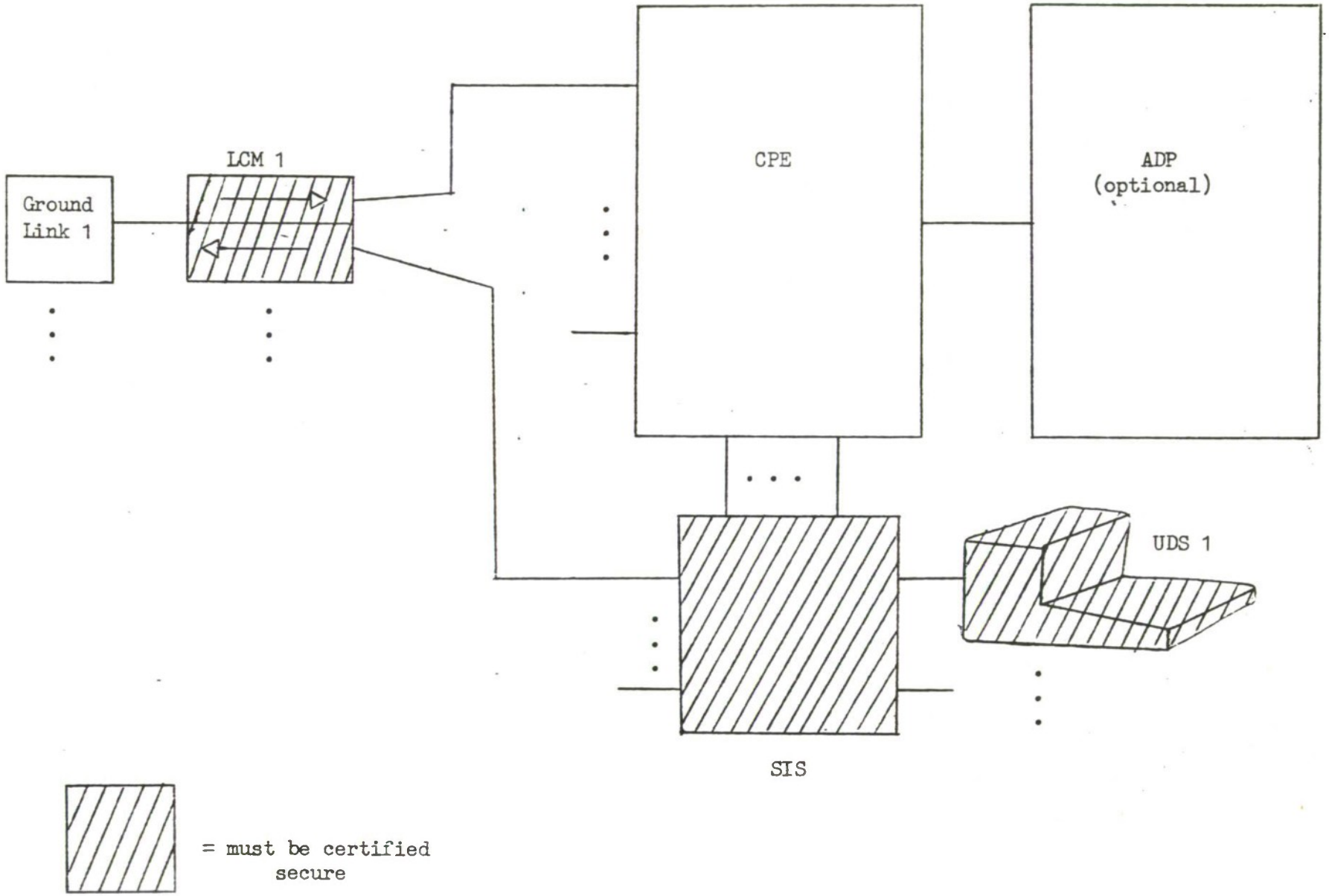


Figure 1. Hardware/Firmware MRT

the LCM.

When the CPE is ready to transmit a message, it deposits the message in the memory of the UDS for the human operator to review for proper classification. The human must logically disconnect the UDS from the CPE and then review both the message content and its format, since message control characters such as STX and ETX can carry encoded information. After ensuring the message is properly classified, the operator presses a button logically connecting the UDS to the LCM. The connect and disconnect functions are handled by the Secure Interlock Switch (SIS). The message passes out the LCM to a cryptographic device and then is broadcast to the ground. Following broadcast, the UDS buffer memory must be zeroed to preclude intermixing of messages.

The LCM, UDS, and SIS are all expected to include microprocessors which are controlled by firmware. Since there is little technical difference between firmware and software (except for the storage media), the firmware in the LCM, SIS, and UDS will require certification, just as software would. Due to the very primitive nature of firmware instructions, certification can be more difficult than for software. If the firmware is performing a simple enough task, this technical certification could be achieved through the so-called "AUTODIN approach". [10]

In the "AUTODIN approach", all the firmware programmers would have to be cleared (or clearable) to the highest level of information processed by the E-4 (Top Secret, SI/SAO, and SIOP-ESI). (2) The firmware would have to be developed in a system high environment and protected from clandestine modification as SI/SAO material. No off the shelf software, such as cross-assemblers, could be used without hand verification of their binary output. Finally, the firmware would have to be subjected to exhaustive testing to assure that bugs were not present. As a major aid to the verification efforts, the design of the firmware should follow the principles of the Bell and LaPadula [1] model for ADP security, properly interpreted for a microprocessor environment, to ensure completeness of implementation.

2.2.2 Kernel in the UDS Implementation

---

(2) [5], Page 3-50

Review of the functions required in the LCM, UDS, and SIS firmware, indicates that they are quite complex for simple microprocessors. The LCM must do a sophisticated job to ensure that the CPE does not send out control signals. Therefore, the LCM must perform error detection/correction functions, a large amount of buffering, and message ACK/NACK protocols. Similarly, the UDS may become an intelligent terminal with sophisticated editing and formating requirements. As the complexity increases, the processing involved approaches that of a small minicomputer. (3) In such a case, the task of certification using the AUTODIN-approach becomes very difficult, primarily due to the very large amount of work required in the exhaustive testing area. The work needed for exhaustive testing goes up very rapidly as complexity increases.

For a somewhat sophisticated UDS, a simpler, less costly, approach to implementing and certifying an intelligent UDS would be to use a security kernel in the UDS to support an MRT function. (See Figure 2). Such a function would be similar to the downgrading terminal used with the security kernel demonstration system currently running on the PDP 11/45 [13]. The minicomputer in the UDS could, for example, be a small Honeywell Level 6 processor with the necessary hardware to support a security kernel [3]. The Level 6 is designed to be ruggedized and to be inexpensive enough to be used in terminal applications. The security kernel for a UDS could be smaller and simpler than the current PDP 11/45 kernel which is itself less than 1000 program statements), because the UDS need not support sophisticated multiprogramming or a file system. The kernel being only a very small portion of the UDS software, significantly reduces (see Section 5) the cost of security implementation and certification. Non-kernel software can be written by uncleared individuals with off-the-shelf development tools.

Use of the kernel technology in the UDS also reduces the overall risk of the MRT approach. A hardware/firmware MRT has never been implemented before, while a kernel-based MRT is currently part of the security kernel demonstration running on the PDP 11/45 at MITRE. Since the feasibility of the security kernel was demonstrated over two years ago, kernel technology is now being offered by industry. The MITRE kernel has also successfully completed major technical design verification, and the verification technology is

---

(3) Indeed, many currently available intelligent terminals have full scale minicomputers in them.

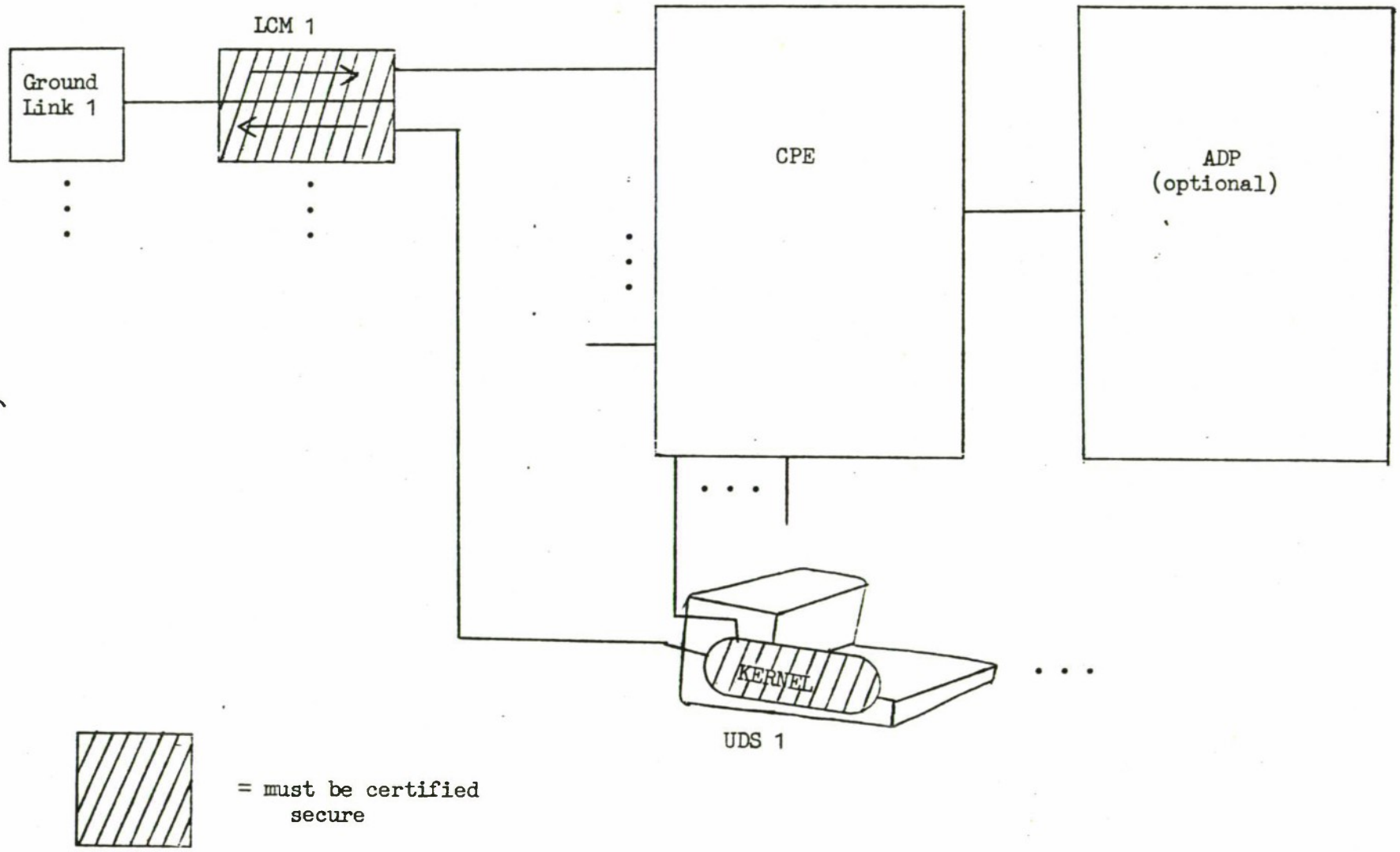


Figure 2. Kernel in UDS

currently assimilated.

2.2.3 MRT Concentrator

Use of a kernel in the UDS simplifies the certification of the UDS and eliminates the need for an SIS. However, that option requires a secure minicomputer in each UDS. Further cost savings can be achieved by moving the the security kernels from each UDS to a single MRT Concentrator minicomputer. (See Figure 3). This MRT concentrator could, for example, be a somewhat larger version of the Honeywell Level 6 processor which could run a security kernel very similar to the MITRE PDP-11/45 kernel [12]. The UDSs could be cheaper dumb terminals, and the secure LCMs could now be replaced by conventional modems.

The MRT Concentrator would work very similarly to the downgrading terminals on the secure PDP 11/45 demonstration. Messages would arrive at the concentrator from the radio receivers, security levels would be marked by the kernel, and the messages would then be turned over to uncertified code to be passed on to the CPE and/or ADP segments. Any information received from the CPE would be treated as system high until it was reviewed by a UDS operator for potential downgrading. The kernel would ensure that no message is downgraded from system high without approval by a human operator. This is the same operation that is currently demonstrated using the system runnong on the PDP-11/45 with a security kernel.

The MRT concentrator can further reduce costs and risks from a hardware/firmware MRT by

1. Eliminating the Secure Interlock Switch,

2. Replacing the complex secure Line Control Modules with simple modems.

3. Reducing the complexity of the User Display Stations.

Of course, the security kernel of the MRT concentrator itself must be certified, but this task will be very similar to the PDP-11/45 kernel certification. Development and certification of a security kernel for a Level 6 minicomputer is currently underway by Honeywell under ESD's engineering development Project 2239. Much of this cost need not be borne by the E-4 program. The hardware cost of an MRT concentrator is estimated to be approximately $75K.

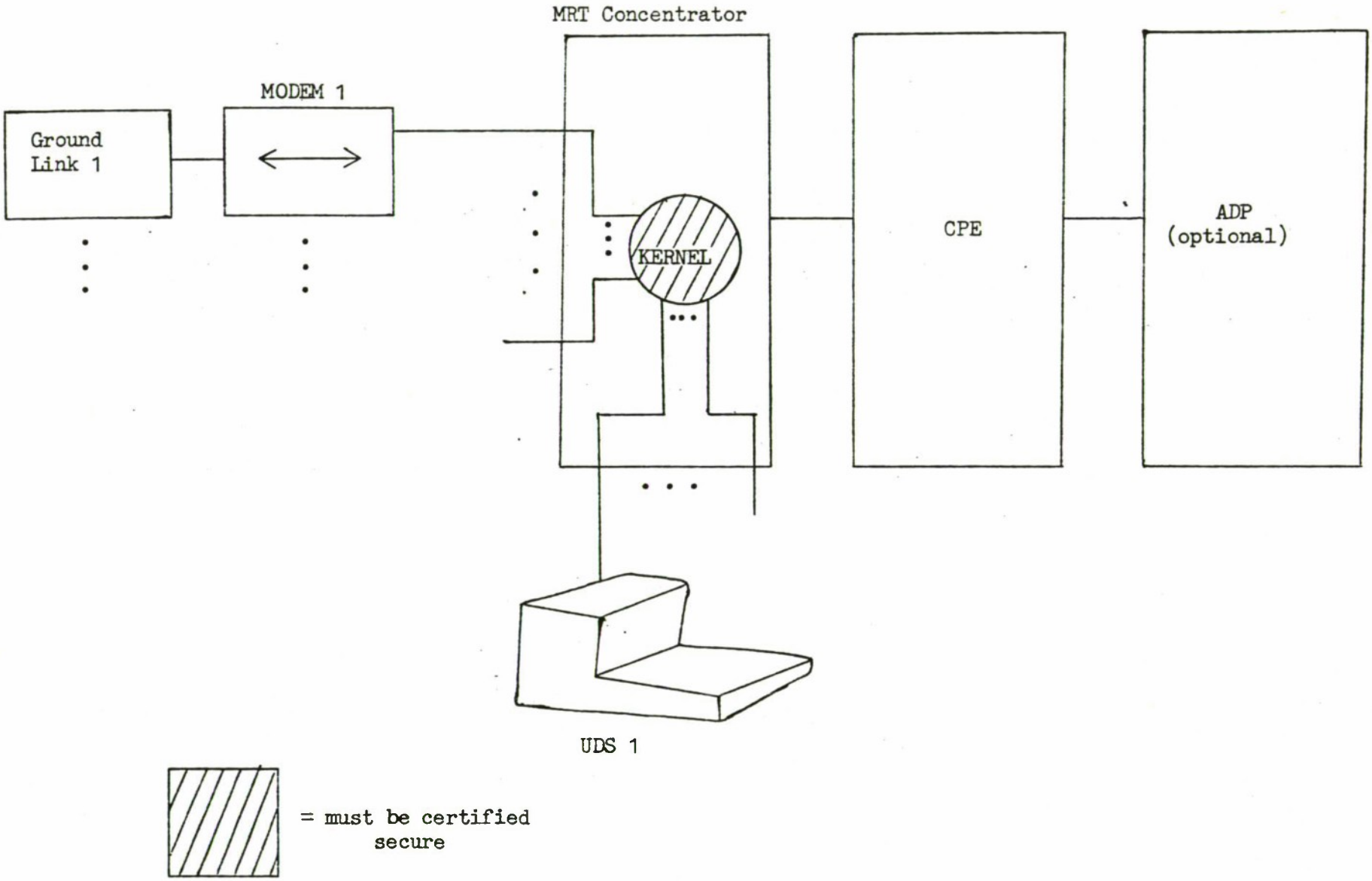


Figure 3. MRT Concentrator

2.3 Human Interface Problems

MRT systems operated in a strictly stand-alone mode suffer from a number of human interface problems. While in theory none of the problems is insurmountable, in practice, some of them may have very undesirable consequences.

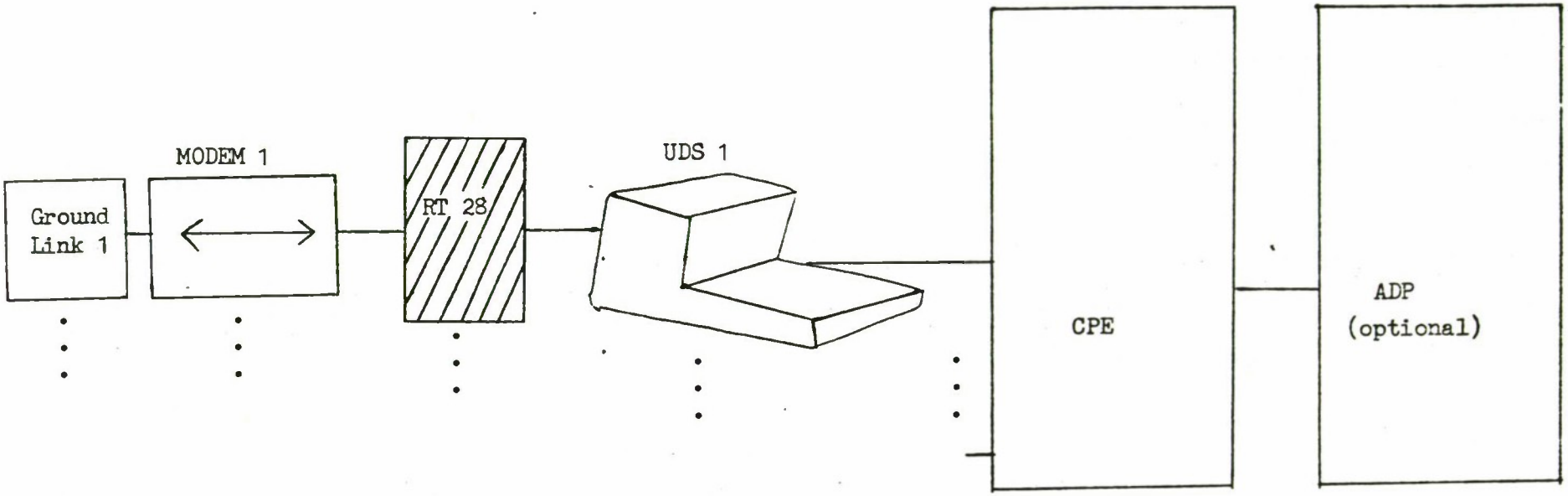
First, the level of message traffic may impose an unreasonable work load on the MRT operator. The SDP states that this is not a problem, based on the number of messages expected and the assumption that each operator performs the MRT function for his/her messages. However, if the number of machine generated messages were to go up significantly, particularly if user requirements dictate that the CPE (or ADP) must generate message acknowledgement messages which must undergo MRT review, then the workload could become operationally unacceptable. On the other hand, if the workload is indeed low, a torn paper tape system using a reperforator transmitter such as the RT 28 or even a manual transcription system may be sufficient, totally eliminating the need for either security kernels or hardware/firmware implementations. (See Figure 4.)

The second human interface problem comes from the fact that reviewing messages for correct security classification tends to be a boring, tedious job. Most, if not all, messages coming from the system will have correct security markings. Experience has shown that the human operator may come to trust the system, and security review may become cursory and approval almost automatic. With the operator thus lulled into a false sense of security, it will become easy for a misclassified message to get by the MRT review due to a single human failure. (4) This problem could be addressed by a two-man review process with the attendant additional workload or by computer support as described in the following sections on the integrated MRT approaches.

The third human interface problem is that it is very hard to identify a security compromise in an outgoing message. Classified information can, for example, be encoded into the low order digits of numbers appearing in outgoing messages. Sequences of ASCII control characters can be used to encode information. For example, if a message is being displayed on a CRT terminal, the ASCII sequence BLANK-BACKSPACE will have no visual effect on the screen. Thus, information could be surreptitiously released

---

(4) Note that we are not assuming the MRT operator to be malicious -- only prone to careless mistakes.



= must be certified secure

Figure 4. Reperforator/Transmitter MRT

by the following algorithm:

a. Encode the desired classified information as a string of binary 0's and 1's. Assume 50% of the bits are 1s.

b. After every 10th character in an outgoing message, if the next bit in the classified information is a zero, do nothing. If it is a one, then insert a SPACE-BACKSPACE pair.

c. The recipient need only reverse the procedure to read the data.

Using the procedure above which inserts on the average of 2 characters in every 20 transmitted (a 10% degradation), the malicious software can release classified information at the rate of 60 bits per second, assuming a 4800 baud transmission rate and 8 bits per character. 60 bits per second is only 20% slower than a single communications channel on the AFSATCOM E-4 communications link. The SPACE-BACKSPACE is only one example of a variety of encoding schemes which can conceal classified information from the MRT operator. This problem of identifying what is classified is inherent in the stand alone MRT and can most effectively be solved by one of the integrated MRT approaches below.

3. MRT INTEGRATED WITH GENERIC LEVEL A

3.1 Overview

In order to overcome the human interface problems and further reduce the costs of the Block II ADP in Generic Level A, this section examines integrating the MRT concept with the Communications Processing Element (CPE). Two basic approaches are outlined -- a controlled environment approach and a certified security kernel approach.

3.2 CPE Controlled Environment

One of the major problems with the stand-alone MRT concept is that a single human failure (due to carelessness - not necessarily maliciousness) can result in a compromise of information. If 99.99% of the messages that a human must review are classified correctly, then the .01% which are not classified correctly may easily escape human detection. If the CPE were operated in a

Controlled Environment mode (5) then the human at the MRT would not be the only security check. A controlled environment requires a minimum clearance level for all users of the system and correctly designed, extensive (but not penetration-proof) security software throughout the system. Appendix A contains sample requirements for a controlled environment communications processor for the E-4. These types of requirements are currently being applied to the SATIN IV program.

The controlled environment in Generic Level A still requires a stand-alone MRT capability. It reduces the impact of human error, however it cannot reduce the stand-alone MRT costs.

3.3 CPE Security Kernel

In section 2.2.3, we examined the possibility of an MRT concentrator processor using a security kernel as a front end to the CPE. One could justifiably ask -- why use a front-end processor to a communications processor which serves as a front-end processor to yet a larger processor? Indeed, a security kernel could be implemented in a CPE to perform both the MRT and CPE functions. (See Figure 5). The security kernel in the PDP 11/45 security kernel demonstration system reflects precisely this approach. Appendix C contains system specification requirements for a security kernel for the E-4. [8] discusses the feasibility of a kernel based communications processor.

Running the MRT terminals with a CPE security kernel gains all of the cost savings and risk reductions of the MRT concentrator and results in additional costs savings by consolidating the MRT and CPE processors. The additional processing load imposed on the CPE by the MRT functions is estimated to be relatively small.

Running a security kernel in the CPE achieves a major advantage over a controlled environment -- the human operator is no longer the primary security control. The security kernel ensures that message classification is correctly marked and maintained, and, because the kernel has been formally proven correct, it can be depended upon to reliably protect classified information. The MRT

---

(5) DoD 5200.28 and AFR 300-8 define the possible modes of operation - Dedicated, System High, Controlled Environment, and true Multi-Level.

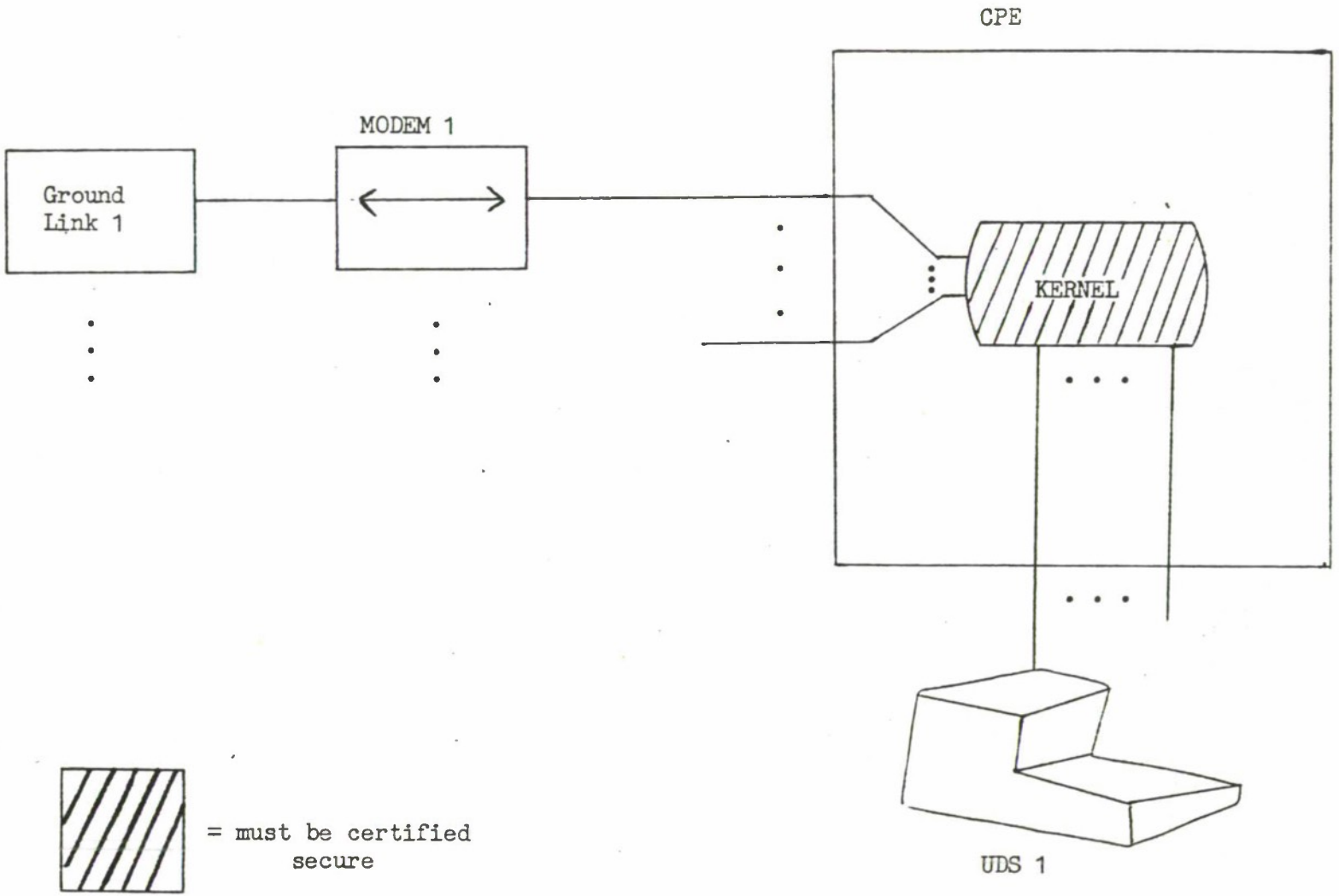


Figure 5. Kernel in CPE/Generic Level A

approach could therefore be abandoned entirely in Generic Level A if a kernel were implemented in the CPE (although it may be kept to support Generic Level D, See section 4.2.1.) NSA, ARPA, and OSD(I) have all formally or informally stated that the security kernel is the most viable currently known approach to achieving multilevel computer security. The technical risk in certifying a security kernel and its software is certainly much lower than in certifying three distinct new (and potentially complex) hardware/firmware devices (the LCM, SIS, and UDS).

The security kernel for a CPE can be implemented on several types of hardware as described below.

3.3.1 Compatible Burroughs D-Machine

The Burrough's D-machine procured in Block I could be reprogrammed (microcode) to provide the features necessary to support a security kernel (segmentation and multiple processor states) while maintaining user mode program compatibility. An effort of this type for the D-machine was described in [4] for emulating the Honeywell 6180. A similar design could be done for the Block I processor. If the D-machine is heavily loaded, it may be desirable for efficiency reasons only to implement the segmented address computation function in a separate D-machine CPU attached to the primary processor. This second processor would have no peripherals and only perform the address translation functions.

3.3.2 Burroughs In-House Kernel

A group at Burroughs led by Dr. George Cowan of the University of Wisconsin is developing a security kernel in-house on the D-machine at the Burroughs plant in Paoli, Pennsylvania. Very little is known about this effort, but presumably it would not be user mode software compatible with the Block I processor. However, such a design with their version of a security kernel could be discussed with Burroughs.

3.3.3 Honeywell Secure Communications Processor

Although the CPE is available from Block I, it is expected in the SDP that much of the Block II software will not be taken from Block I. Because of the high cost of software and the high cost of remicroprogramming compared to current hardware costs for a CPE, it becomes reasonable to consider replacing the D-machine. In

particular, Honeywell's currently announced Level 6 line includes a Secure Communications Processor (SCOMP) which is being developed under ESD Project 2239 to specifically support a security kernel. The SCOMP need only be airborne qualified: the contractor has indicated that it may be nearly qualifiable already. (The MCI contract for the SCOMP only calls for militarization design for a ground environment.)

The SCOMP, in addition to its security capability, has several non-security related advantages over the D-machine. First, its instruction speed is faster. Second, it has a unique feature called a multi-line controller for very efficient communications line handling. Third, the kernel will provide the necessary file system functions which are not available from the Block I software. Fourth, a SCOMP with new technology is expected to be significantly less costly than the D-machine.

3.3.4 Other Communications Processors

Other militarized communications processors are or are expected to be available soon which can support security kernels. These include, among others, the ROLM 1602 with secure executive mode (specifically designed for implementing security kernel technology), the UNIVAC AN/UYK-20 with special security hardware, and the new militarized PDP 11/45 and 11/70 from the Norden division of United Technologies. Any of these processors could be used to implement a kernel for the CPE.

4. MRT INTEGRATED WITH GENERIC LEVEL D

4.1 Overview

Generic Level D introduces a significant new complexity into the Block II ADP program - a large general purpose interactive system. Integration of the MRT concept using the security kernel to reduce cost and risk again seems to present the best options for achieving security. Generic Level D is being considered with two major approaches -- a Honeywell WWMCCS compatible approach and a non-Honeywell approach. Security alternatives of each of those approaches will be broken down into two options - controlled environment and security kernel.

4.2 Honeywell WWMCCS Compatible Approach

Software compatibility with the present WWMCCS Honeywell 6000 processors can provide two major advantages to the E-4 program. First, ground applications can be run on the E-4 with little or no modification; and second, applications development can be done on existing ground based ADP systems, minimizing investment in support facility hardware and software. In addition, programmer training costs can be reduced by exploiting the commonality of architectures.

Honeywell has proposed a so-called "Medium 6 (M6)" processor for the E-4 Block II ADP. The M6 is upwards compatible with the 6000 WWMCCS GCOS, but does not have a multilevel security capability when used with GCOS III software. Honeywell has proposed to the WWMCCS community the development of a secure GCOS IV, but that effort has extremely high risk and cost comparable to a completely new large scale operating system development. At this time, the WWMCCS Program Office is not aggressively pursuing GCOS IV.

However, Honeywell can also provide a version of the M6 with upwards compatibility with the 6180 Multics system. Honeywell has informally indicated that the differences between a GCOS M6 processor and a Multics M6 processor are only microcode changes. The microcode changes would reduce processor speed somewhat but board changes (20% or fewer) would be required only to achieve maximum processing power.

Multics is a broadly capable and currently operational operating system with very advanced security capabilities. In addition, it is inherently upwards compatible with GCOS. Multics is presently installed at a number of commercial and military sites around the country and overseas. In particular, the Air Force Data Services Center (AFDSC) in the Pentagon and the Rome Air Development Center (RADC) both operate the Multics system on Honeywell 6180's.

4.2.1 Multics Controlled Environment

The Air Force Data Services Center (AFDSC) is presently operating Multics designed for a controlled environment [15]. The commercial Multics security controls were enhanced to provide 8 security levels with 16 categories. All users of the AFDSC system are cleared at least to the SECRET level and the Multics security

controls are used to separate SECRET and TOP SECRET users and information. Processing of compartmented information is not currently required. A manual review approach is used to control output below SECRET from the system: all outputs are protected as either SECRET or TOP SECRET until reviewed for proper classification. GCOS compatibility is provided as described below in Section 4.2.3.

A controlled environment Multics could be run on the M6 airborne qualified processor with minimal cost and risk. The vast bulk of an interactive operating system needed to support the full range of E-4 requirements is already in operational use at AFDSC. Multics is designed to easily accommodate a wide variety of new terminals. The principle changes necessary would be for unique airborne qualified peripheral devices. These same type of software changes would be required for a GCOS version.

Honeywell had initiated development of a security kernel based SCOMP under Project 2239 (terminated by AFSC in August 1976) to be the new Multics front-end processor, replacing the current Datanet 355. If completed, such a kernel based SCOMP could easily function as the CPE for Block II and implement an MRT concentrator function. (See Figure 6). Thus, outgoing messages from the non-kernel Multics controlled environment would be reviewed by an MRT process in the CPE. (See section 3.3.3.)

The controlled environment Multics with a kernel-based SCOMP front-end can provide a low risk security approach to Generic Level D. Most security software is off-the-shelf (viz., user controls). In addition, it is supported by a security kernel based MRT system in the SCOMP front end. This configuration also provides a security growth capability to a Multics security kernel as described in Section 4.2.2 below.

4.2.2 Multics Security Kernel

Under Project 2239 (terminated by AFSC in August 1976), ESD had begun developing a security kernel for Multics which would be upwards compatible with the controlled environment Multics. If these developments were completed, when this kernel became available, it could be retrofitted into the E-4 to provide certifiable internal security controls. (See Figure 7). At this point, the MRT procedures would no longer be needed as the primary security controls and could be removed. In addition, because the controlled environment Multics provides acceptable security using the MRT approach in a

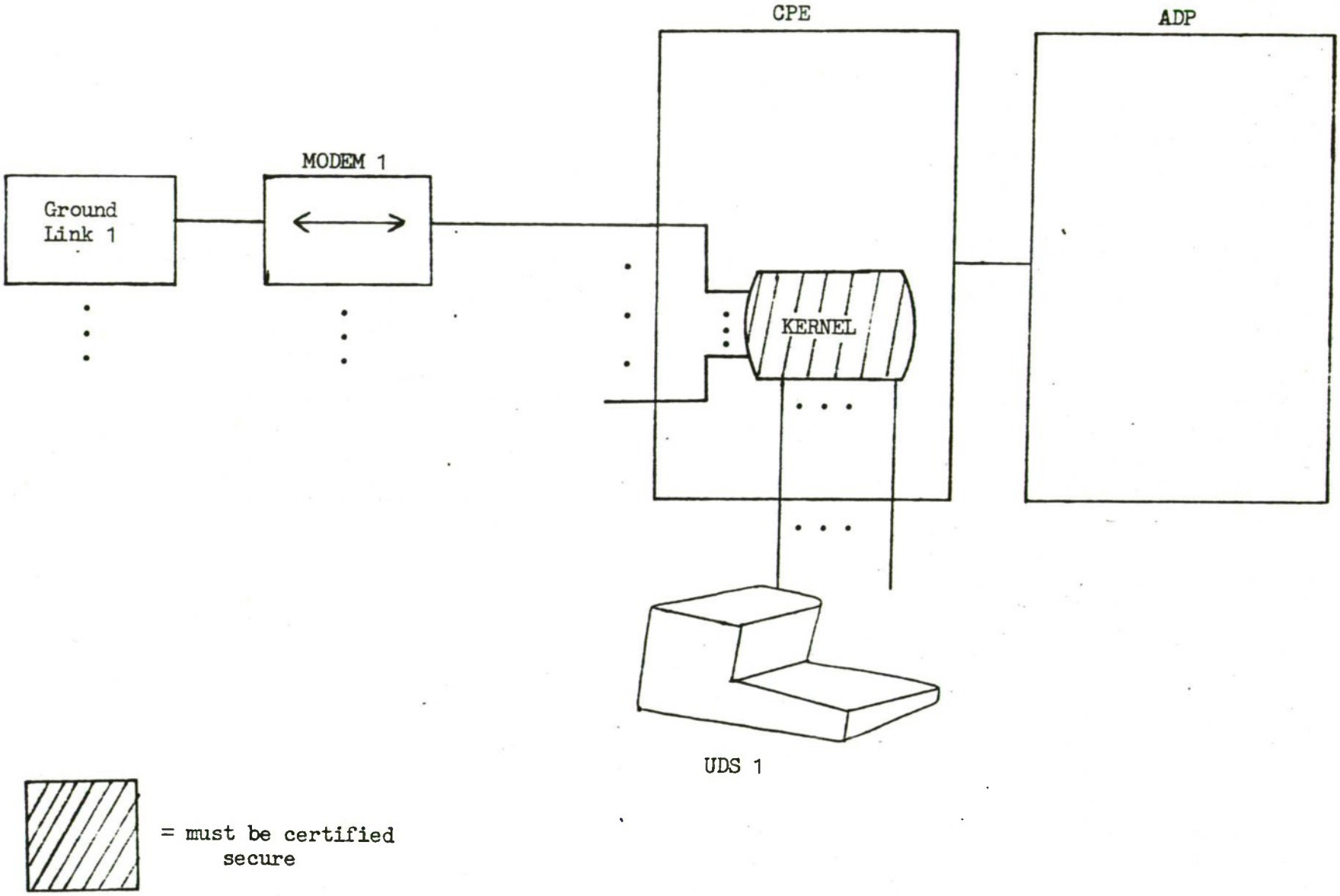


Figure 6. Controlled Environment MRT/Generic Level D

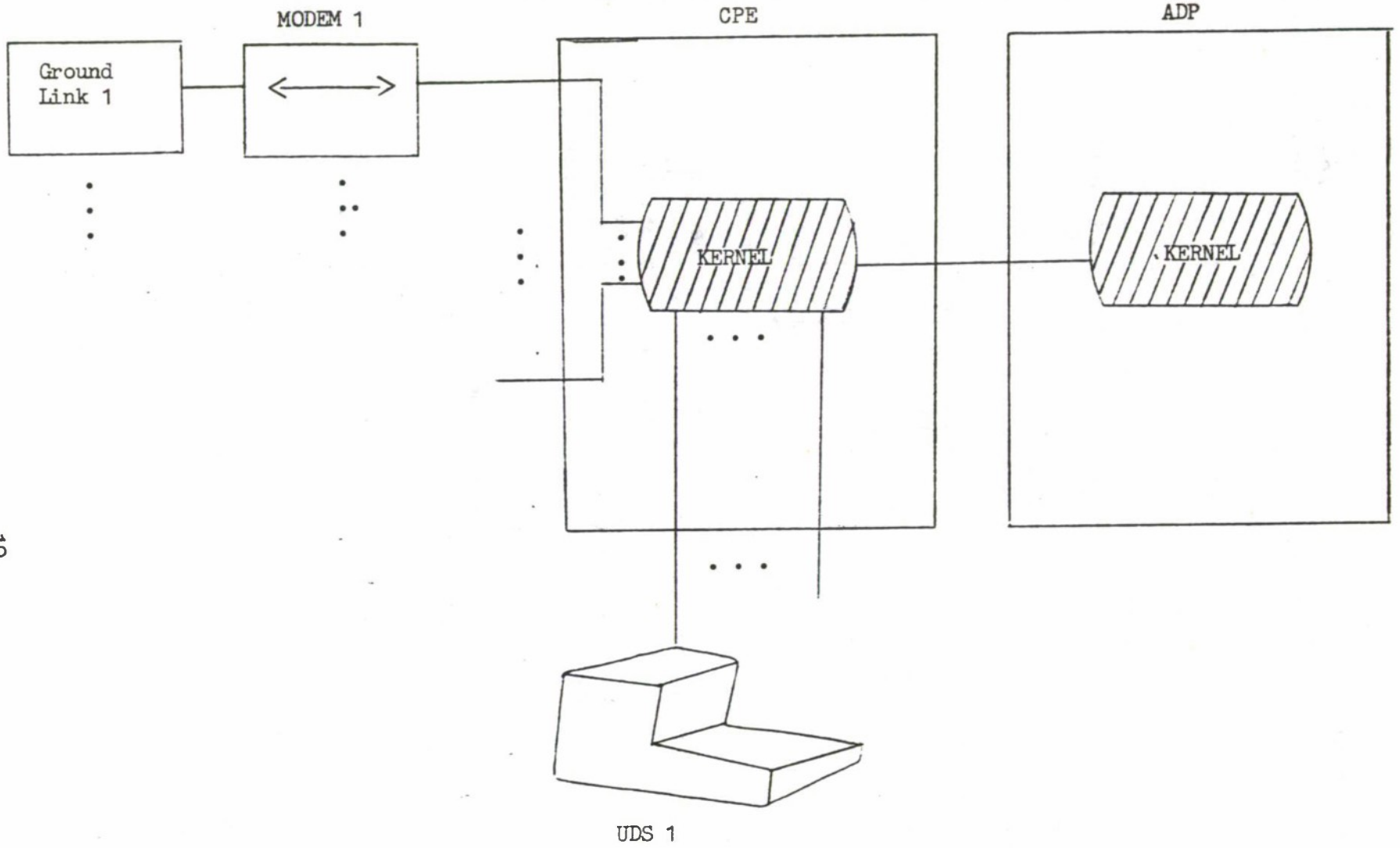


Figure 7. Kernel/Generic Level D

kernel based front-end processor using the Honeywell Level 6 SCOMP, this approach fails "soft" if the Multics kernel is not approved for multilevel operations. (6) The Multics kernel effort is described in [9] and [2].

4.2.3 Multics - GCOS Compatibility

Multics presently has a capability called GCOS encapsulation which provides capability to run GCOS slave mode programs and compilers. GCOS encapsulation is in use at RADC (and to some extent at AFDSC) with good success. RADC, for example, has just brought up their JOCIT JOVIAL compiler under GCOS encapsulation to provide language level WWMCCS compatibility.

Programs run under GCOS encapsulation are completely subject to the Multics security controls. Any GCOS supervisor calls (MME instructions) are intercepted by a user mode Multics program and translated into calls on the Multics operating system. GCOS encapsulation, however, only provides compatibility for slave mode programs. Master mode programs such as SAC's SONIC system or the GCOS supervisor itself cannot run under GCOS encapsulation. GCOS encapsulation is described in [7].

Although GCOS master mode programs cannot be run under GCOS encapsulation, they are generally not needed. Native mode Multics already contains the major functional capabilities that were implemented in SONIC to correct deficiencies of the much older GCOS. For example, native mode Multics routines can easily access terminals, process real time interrupts, and share areas of main memory outside the operating system. This ability to share memory can significantly reduce the Block II main memory requirements, since many programs that were planned to have several copies in core at once can be shared with only one copy under Multics. (7)

Both the controlled environment and kernel based Multics systems can provide a substantial degree of WWMCCS

---

(6) Lack of certification is unlikely to result from technical failure; however, there is little experience in the policy issues in in the area of hardware/software certification.

(7) If complete GCOS compatibility is a requirement, RADC has proposed a task to develop a Virtual Machine (VM) GCOS which can run master mode programs.

GCOS slave mode compatibility with existing off-the-shelf software. A growth capability also exists for hardware level compatibility for master mode programs in the GCOS supervisor, but such a requirement seems unlikely for the E-4 program.

4.2.4 Non-Security Related Multics Features

Multics was developed by MIT, GE, and Bell Labs under ARPA sponsorship to be an advanced interactive and batch system. As such, it has a number of features which could reduce costs for the E-4 program.

Multics uses a virtual memory system to allocate main memory. Users programs are broken up into 1024 word pages, and only those pages which are needed at a given time are present in main memory. Paging results in much more efficient use of main memory, since programs typically use only small amounts of memory at any one instant. Paging can also significantly reduce disk channel traffic by swapping only those pages which are needed, when they are needed, rather than swapping entire core images. Moreover, only pages that have been modified are swapped back to the disk. For example, GCOS systems at AFDSC often come close to disk channel saturation. However, Multics at AFDSC, which has a larger mass memory, does not come close to channel saturation.

Multics also provides a very sophisticated interactive program development environment. Programmers have available powerful interactive text editors, higher level language interactive debugging tools, documentation aids, and configuration control aids. Honeywell, for example, uses Multics at their Billerica, Mass. plant as a "software factory" producing software for other Honeywell computers. Their experience has been more than a 50% reduction in software development costs using Multics. This reduced software development cost can reduce both ESD contract costs and the cost of ground based software development by the users.

The Multics operating system is coded almost entirely in a higher order language PL/I. There will be much lower costs for any modifications necessary for E-4 peripherals than the corresponding modifications in an assembly language operating system like GCOS.

This is, of course, only a brief summary of Multics features. More information can be found in the Multics Users Guide [6].

4.3 Non-Honeywell Approach

The non-Honeywell approach assumes that WWMCCS compatibility is not a major requirement and that a competitive procurement is desired. A non-Honeywell approach can be interoperable with WWMCCS without being software compatible. The SDP expressed a major concern over word lengths -- 16 vs 32 vs 36 bits. The ARPANET experience indicates that systems of widely varying word lengths and software characteristics can be made interoperable, although not software compatible.

4.3.1 Controlled Environment

In theory, a controlled environment similar to that developed for the AFDSC Multics could be developed for other systems. However, no such controlled environment has as yet been built. The cost and risk of development is very much a function of the type of base operating system that is used. For example, a controlled environment could be implemented on IBM's VM/370 without too much difficulty. However, a base of OS/360/370 would be almost impossible to upgrade to a controlled environment. Inputs to a system specification for a controlled environment may be found in Appendix B.

4.3.2 Security Kernel for a non-Multics General Purpose Interactive System

Given that the Multics security kernel effort is the first such effort for a large general purpose time sharing system and that Multics is currently the best suited such system, an attempt at this time to build a kernel for another such large system must be considered a R&D task with substantial risk. Therefore, this option is not recommended for the E-4.

4.3.3 Small Interactive System Approach

The requirement stated in the SDP for the ADP element of Block II does not require as large a processing capacity as a Honeywell M6 processor. Only 18-24 terminals are planned without any major CPU usage programs. Therefore, a modern small scale time sharing system may be more cost effective than a large scale general purpose system.

One such small scale system is the UNIX system [11] developed at Bell Labs for the PDP-11/45 and 11/70. UNIX supports about 25-30 terminals with a fairly sophisticated

system based in part on the Multics design, but scaled down to the 11/45.

MITRE is presently modifying the PDP-11/45 security kernel to run UNIX [14]. Expected completion of an initial non-production version is CY1976. Such a kernel based UNIX could run on a militarized 11/45 or 11/70, as recently announced by the Norden Division of United Technologies. A UNIX with a security kernel could be provided for Generic Level D at much lower cost and risk than a large scale general purpose kernel.

However, the PDP-11/70 may not be large and fast enough to support Generic Level D. In this case, a UNIX-like system could be implemented on a Honeywell Level 6 SCOMP with a security kernel. The largest Level 6 processors are expected to be in the scale of the 11/70, but the Level 6 kernel will support multiprocessing which the 11/70 kernel will not.

Similar to the Multics kernel option, UNIX or UNIX-like kernels on 11/70's or Level 6's have a fail "soft" capability if the kernel is not certified. In the event of failure, the system degrades to a controlled environment with an MRT system in the front end processor.

5. COST AND SCHEDULE ESTIMATES

OPTION	Δ FOR SECURITY	Δ FOR OTHER COST	TOTAL COST Δ	Δ MONTHS
Hardware/Firmware MRT w/o Certification	0/0	0/0	0/0	0
Hardware/Firmware MRT with Certification	+600K/0	0/0	+600K/0	+12
Generic Level A Reperforator-Transmitter	0/+8K	-1200K/-320K	-1200K/-312K	0
Generic Level D Reperforator-Transmitter	0/+8K	-1400K/-430K	-1400K/-422K	0
Generic Level A Kernel in UDS	+250K/+5K	-1000K/-270K	-750K/-265K	+6
Generic Level D Kernel in UDS	+250K/+5K	-1200K/-380K	-950K/-375K	+0
Generic Level A MRT Concentrator	+1000K/+75K	-1200K/-320K	-200K/-245K	+6
Generic Level D MRT Concentrator	+1000K/+75K	-1400K/-430K	-400K/-355K	0
Generic Level A Controlled Environment	+1000K/0	0/0	+1000K/0	+6
Generic Level A with CPE Kernel				
Segmented D-machine	1000K/ 50K	-1200K/-320K	-200K/-270K	12
Burroughs CPE Upgrade	3000K/ 150K	-1200K/-320K	1800K/-170K	18
HIS Level 6	500K/ 150K	-1200K/-320K	-700K/-170K	12
Other SCOMP	3000K/ 150K	-1200K/-320K	1800K/-170K	18

OPTION	Δ FOR SECURITY	Δ FOR OTHER COST	TOTAL COST Δ	Δ MONTHS
Generic Level D Honeywell Approach				
Multics Controlled Environment	+500K/+300K	-6400K/-430K	-5900K/-130K	-12
Multics Kernel (See Note 1)	+1500K/+300K	-6400K/-430K	-4900K/-130K	+12
Generic Level D Non-Honeywell Approach				
Controlled Environment	+14000K/+200K	-1400K/-430K	12600K/-230K	+24
Kernel	28000K/ 200K	-1400K/-430K	26600K/-230K	48

Note 1: If GCOS VMM is desired instead of GCOS encapsulation, add these figures to Generic Level D, Honeywell Approach.

1500K/ 50K	500K/0	1000K/ 50K	0

All costs assume ESD Project 2239.

Each cost is formatted as: Development Cost/Per Copy Cost.

6. CONCLUSIONS

For Generic Level A, the best approach seems to be an MRT system implemented with a security kernel in the CPE. This approach reduces LCM and UDS costs, eliminates the SIS hardware, and reduces the risk of certifying the new hardware/firmware MRT approach by using the existing downgrading system technology.

For Generic Level D, the best approach seems to be a controlled environment Multics processor running the AFDSC security controls. This system is available off-the-shelf and provides growth to the Multics security kernel when it is available. The Multics should have a Honeywell Level 6 SCOMP front end processor, with a security kernel supported MRT. This approach provides significant WWMCCS applications compatibility, security with the MRT system, and growth capacity for both security and full WWMCCS compatibility.

7. REFERENCES

[1] Bell, D.E., and L.J. LaPadula, "Secure Computer System: Unified Exposition and Multics Interpretation", ESD-TR-75-306, March 1976.

[2] Biba, K.J., et al, "A Preliminary Specification of a Multics Security Kernel", MITRE WP-20119, April 1975.

[3] Broadbridge, R., and J. Mekota, "Secure Communications Processor Specification", ESD-TR-76-351, Vol II, June 1976.

[4] Burke, E.L., et al, "Emulating a Honeywell 6180 Computer System", RADC-TR-74-137, June 1974.

[5] ESD/YS, "System Development Plan for E-4 AABNCP Automatic Data Processing" 12 January 1976.

[6] Honeywell Information Systems, Inc., "Multics Users' Guide", Order No. AL40.

[7] Honeywell Information Systems, Inc., "Multics CCOS Environment Simulator", Order No. AN05.

[8] Information Systems Technology Applications Office, "The Feasibility of a Secure Communications Executive for a Communications System", MCI-75-10, August 1974.

[9] Information Systems Technology Applications Office, "ESD 1974 Computer Security Developments Summary", MCI-75-1, December 1974.

[10] Lipner, S.B., "SATIN Computer Security", MCI-75-2, September 1972.

[11] Ritchie, D.M., and K. Thompson, "The UNIX Time-Sharing System", Communications of the ACM, Vol. 17, No. 7, July 1974.

[12] Schiller, W.L., "The Design and Specification of a Security Kernel for the PDP 11/45," ESD-TR-75-69, May 1975.

[13] Stork, D.F., "Downgrading in a Secure Multilevel Computer System: The Formulary Concept", ESD-TR-75-62, May 1975.

[14] White, J.C.C., "A UNIX Executive for Use with the PDP-11/45 Security Kernel", MITRE WP-20056, 5 December

1974.

[15] Whitmore, J., et al, "Design for Multics Security Enhancements", ESD-TR-74-156, December 1973.

APPENDIX A

CPE CONTROLLED ENVIRONMENT SPECIFICATION

E-4 Block II CPE Security

1 General. This section provides additional security requirements for E-4 Block II communication processor elements (CPEs). The security requirements of this section, in conjunction with the other security-related requirements of this specification, represent the total set of functional requirements which must be met by CPEs for countering threats to the system. The system shall be a multilevel secure system. Specifically, the system will handle messages with various security classifications and will support subscribers with various security clearances; therefore, there will be subscribers which must be prevented from gaining access to classified information for which they are not cleared. This appendix supplements other security-related requirements of this specification in specifying the internal security controls necessary to prevent security compromise of classified information (as defined in DoD 5200.1-R, paragraph 1-305) in a multilevel system. In addition, this appendix specifies requirements necessary to supplement the integrity-related requirements of the specification when such requirements have security implications.

1.1 Applicable definitions. The following definitions are an aid to better understanding the functional requirements specified in paragraph 3.

a. Subject. A subject is an active entity that accesses information (e.g., in a conventional communications processor, a subject is an internal process or I/O channel).

b. Object. An object is an entity that holds information and is acted upon by a subject (e.g., a buffer, file, etc.)

c. Intermixing. The processors must be able to detect the inadvertent appending of complete or partial messages to a preceeding message. If this condition is caused within a processor, it is called intermixing or merging; if it is detected on receipt by a processor, it is called a straggler.

d. Trusted Process. A process (i.e., a program in execution - one type of subject) may be designated trusted if it has been both verified analytically and demonstrated to show that its operations will not cause or allow

information of a given classification and access categories to be transferred into objects of lower classification or different access categories, designated untrusted. This definition of "trusted" also applies to all types of subjects.

2. Applicable documents. The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the referenced paragraphs of the documents referenced herein and this specification, the contents of the referenced paragraphs of the referenced documents shall be considered a superseding requirement.

DoD 5200.1-R Information Security Program Regulation (Nov 73).

3 Requirements. The security of classified information internal to all CPEs shall be maintained in accordance with the provisions of DoD 5200.1-R (para.1-402, 6-100, 7-100, 7-103) and the additional provisions prescribed by this specification regarding the safeguarding of classified information. The following security requirements shall be maintained during both normal and degraded operation.

3.1 User autherntication. User authentication shall be provided to properly and uniquely identify valid users and to prohibit illegal access to the system. Any illegal attempt to access the system shall be denied and shall be reported.

3.2 Message security validation. Each message shall be validated for security.

3.3 Straggler and intermixing protection.

3.3.1 Straggler and intermixed protection at all processor levels.

a. Recognition of stragglers on MESSAGE receipt at a processor shall be provided.

b. Message intermixing (merging) shall be prevented.

3.4 Authorization and validation for recall of messages and data. (Reference: SDP section 2.2.2.) A message requesting a message or data retrieval shall be validated (as any other message would be) by the system. If the requesting message is validated, a further check shall

then be made for authorization based on security clearance and access category. The requested message or data shall also be examined for deliverability to the requesting subscriber, where deliverability demands that the requestor be the originator of the original message or thatthe requestor as an addressee. If either the authorization or deliverability checks described above should fail, the message or data request shall be reported as an attempted security violation. Delivery of the requested message shall be withheld. The requestor shall be advised of the error in his request.

3.5 Access control. CPEs shall be implemented with trusted processes and an internal access control mechanism (IACM) that shall prevent by positive action misrouting of message that would lead to a security compromise.

3.5.1 Access rules. The policies of DoD 5200.1-R as referenced above shall be implemented by the IACM in all processors in terms of rules that control the access of subjects to objects within these processors. Every object (see 1.1b) shall be explicitly assigned a security attribute that shall represent the classification and categories of the object. Every subject (see 1.1a) shall be explicitly assigned a security access level that shall represent the clearance level and the set of access categories authorized for the subject. The attributes are assigned from the list of classfification and security categories specified for the system in (?). The security categories shall be considered disjoint sets. Within each category, T, S, C. R, U represent a hierarchy. In addition, every subject shall be explicitly designated as trusted or untrusted (see 1.1.d). The rules that control the access of subjects to objects shall be as follows:

a. Before granting information-access, a subject's clearance level and access category set shall be compared with an information object's classification level and categories. the subject shall be allowed to read the element of information only if the clearance level of the subject is greater than or equal to the classification level of the object, and the subject is authorized access to the set of categories that are assigned to the object.

b. No untrusted subject shall have read-access to an object of a higher classification than one to which the subject concurrently has write-access; i.e., information

from an object with a given classification may only be transferred into an object with an equal or higher classification.

3.5.2 IACM and trusted process characteristics. The functional requirements of the following subparagrapsh shall be met by the IACM and trusted processes where designated.

3.5.2.1 Internal modification protection. the IACM shall be protected from modification by programs in execution external to the control mechanism. The IACM shall protect the code of trusted processes from modification. If an attempted modification to the IACM or trusted process code is detected, the program making the attempt shall be aborted and the local operator shall be notified.

3.5.2.2. External modification protection. The IACM and programs implemented as part of trusted processes shall be protected from modification by processor operators; no console action (other then loading) shall cause any modifications to these programs.

3.5.2.3 Execution protection. The IACM shall prohibit the transfer of control to memory areas not designated as executable (e.g., the IACM shall prohibit the execution of code introduced in message form from a subscriber).

3.5.2.4 Universal mediation. The IACM shall mediate (per 3.5.1) all requests made by subjects to access objects; i.e., the IACM shall always be invoked.

3.5.2.5 Physical resource contorol. The IACM shall control all processor hardware features that permit access to the machine's physical resources (e.g., memory, I/O channels).

3.5.2.6 Trusted processes. In addition, the following shall apply to those trusted subjects which are trusted processes:

a. The specific function of a designated trusted process shall always be performed by that trusted process only; i.e., the designated trusted process shall always be invoked.

b. No trusted process shall invoke or otherwise make use

of any untrusted process or rely on information provided solely by an untrusted process.

3.6 Software loading procedures. A validated automatic sequence of operations for loading and initializing the system, including the IACM in the communications processors, shall be provided. Such a sequence of operations shall also validate the hardware functional capabilities of the processors, including the portion of the internal access control mechanism that is embedded in the processor hardware (e.g., using hardware diagnostic checks).

3.6.1 Hardware loading. The bootstrap loader shall be capable of being loaded directly by hardware. The operator shall load the system by performing a single action. No other operator action shall be involved in a successful loading process.

3.6.2 Bootstrap loading. The bootstrap storage medium shall contain a validated loader routine followed by the IACM code and its related data bases. This loader shall be capable of loading the IACM and its related data bases without the assistance of any software other than that contained in the loader. The bootstrap loader shall include a routine to check for proper loading of the IACM code and its related data bases (e.g., a check-sum). It shall be executed immediately upon completion of the loading of the IACM code and its related data bases. In the event any error is detected, the checking routines shall inform the operator immediately of the nature of the error and halt the processor. Upon successful completion of the checking routines, the operator shall be notified and control shall be passed to the IACM for initialization of the system.

3.6.3 Initialization. The IACM shall be responsible for the correct initializing of its own data base as well as the hardware and software security-related functions. The remainder of the system's software shall be loaded under the control of the IACM.

3.7 Security-related hardware and software monitoring. Monitoring of software and hardware operations shall be performed to permit diagnosis of system change due to modification or failure. The software that implements the on-line diagnostic functions shall execute under complete control of the IACM. Specifically, the internal security policy enforced by the IACM and other security-related

functions shall not be relaxed while on-line diagnositc functions are executing, nor during any degraded conditions detected by the diagnostic functions. Software malfunctioning shall cause the attempted generation of a service message relating the nature of the failure followed by a system restart or reload based on the nature of the failure. Hardware malfunctions that jeopardize the operation of the IACM shall cause a system halt, with an indication to the local operator of the cause of the halt. The IACM shall prevent diagnostic functions from modifying the IACM software.

4 Quality Assurance provisions. Verifications shall be performed to determine that the security requirements of paragraph 3 are met. These shall include, but not be limited to, the following.

4.1 User authentication. Proper user authentication shall be demonstrated by users attempting to gain access to a CPE. Verifications involving illegal subscribers attempting to gain access shall also be conducted. These verifications shall insure that all valid attempts are properly completed and all invalid attempts are not completed and are properly reported. Verifications shall also show that no "handshake" or control signal conventions invalidate the security controls of the flow of information.

4.2 Message security validation. The following demonstrations shall be conducted to assure proper security-restricted handling of traffic. Message headers shall bear the various classification markings, but the text messages shall be unclassified.

a. Users with various clearances and access categories shall attempt to enter messages of higher classifications and unatuthorized categories into the system. These demonstrations shall insure that these messages are rejected and notifications of the attempted violations are properly reported.

b. Messages routed to users of lower clearances and/or excluded access categories relative to the messages shall be entered into the system. These demonstrations shall insure that the messages are properly serviced based on the attempted violations.

4.3 Straggler and intermixing protection. The following demonstrations shall be performed to assure proper protection against straggler and intermixed messages in

accordancw tith 3.3.

a. Several messages and partial messages, purposely composed with stragglers attached, shall be input to the system to insure that the processor recognizes the errors, rejects the messages, and properly reports the attempted violations.

b. Several messages shall be introduced concurrently to the processor and directed to the same output device (e.g., UDT). These demonstrations shall insure that proper message separation and framing are performed for the device.

c. Demonstrations shall be conducted in which a message with an overriding precedence shall be directed to a device that is processing a lower precedence message. These demonstrations shall insure that processing of the messages is performed without any intermixing of the messages.

d. Several messages shall be introduced concurrently to the communications processor and directed to the same subscriber. These demonstrations shall insure that proper message separation and framing are performed for the subscriber.

4.4 Message and data recall. The following demonstrations shall be performed to assure that the requirements of 3.4 are met in properly validating and authorizing recall requests. a. Invalidly formatted recall requests shall be introduced to the system. These demonstrations shall insure that the processor handling the requests properly identifies the improper formats, does not perform the message recalls, and sends service messages to the originators stating the nature of the errors. b. Demonstrations shall be performed to insure that attempted recall of messages and data that the requestors are not authorized to receive and properly recognized, that service messages are sent, and that the requestor receives an indication that the request was not accepted.

4.5 Access control. Analyses, demonstration, and tests shall be conducted as necessary to verify that the design and implementation of the IACM and trusted processes in the CPEs meet the requirements of 3.5. Operating tests and demonstrations of the effectiveness of the IACM and trusted processes in meeting the requirements of this appendix shall be conducted, including test procedures

that attempt to violate the IACM rules. These analyses, demonstrations, and tests shall include, but not be limited to, the following.

4.5.1 Access rules. Verification shall be performed to show that the access rules of 3.5.1 are not violated.

4.5.2 Internal modification. Verifications shall be performed to show that a program outside the IACM attempting to modify the IACM or trusted process code is prohibited from doing so, and that the local operator is notified of the attempt.

4.5.3 External modification. Verifications shall be performed to show that console actions do not modify the IACM, its data bases, or programs implemented as part of trusted processes.

4.5.4 Execution protection. Verifications shall be performed to show that any attempt by a program to transfer control to any memory area not designated as executable will be disallowed. Specifically, demonstrations shall be conducted to verify that no actions at a message terminal cause information entered from the terminal to be acted upon as code.

4.5.5 Universal mediation. Verifications shall be performed to show that all attempts made by subjects to access objects are mediated by the IACM.

4.5.6 Physical resource control. Verifications shall be performed to show that all processor hardware features that permit access to the machine's physical resources are controlled by the IACM, and that any program outside the IACM attempting direct (unmediated) access to these resources is denied access.

4.5.7 Trusted processes. Verifications shall be performed to show that trusted processes do not commit unauthorized downgrading of classified information per 1.1.d. Verifications shall also be performed to show that the function of each trusted process is always performed by only that designated trusted process; i.e., that the desingated trusted process is always invoked. Verifications shall also show that no trusted process uses any program of an untrusted process as a suboroutine.

4.6 Software loading procedures. Verifications shall be conducted to assure that the software loading procedures

shall prevent security compromise and operate in accordance with 3.6. These verifications shall be performed to validate the sequence of operations required by 3.6 for loading and initializing the system, including the IACM and its related data bases. These verifications shall also assure that the hardware logical functional capabilities of the processors for accommodating the IACM are validated. These verifications shall include, but not be limited to the following.

4.6.1 Hardware loading. Demonstrations shall be performed to show that the operator can load the system by performing a single action, and that the bootstrap loader is loaded directly by hardware.

4.6.2 Bootstrap loading. Verifications shall be performed to show that any security-related hardware malfunction detected during loading will cause the local operator to be informed of the condition and the processor to be halted. Demonstrations shall be performed to show that when the IACM is not currectly loaded, the condition will be detected, the operator will be notified, and the processor halted. Demonstrations shall be performed to show that no software other than the loader is necessary to perform the loading procedure, and that upon successful completion, the operator is notified and control is passed to the IACM.

4.6.3 Initialization. verifications shall be performed to show that the IACM correctly initializes its own data base and all hardware and software security-related functions. Demonstrations shall be performed to show that all remaining system software is loaded under control of the IACM.

4.7 Security-related hardware and software monitoring. Verifications shall be performed to show that on-line diagnostic routines are executed under control of the IACM, and that any attempt by a diagnostic routine to violate the rules of the IACM is disallowed. Verifications shall be performed to show that hardware malfunctions that jeopardize the operation of the IACM will be detected, the operator will be notified of the condition, and appropriate action will be taken. Verifications shall also show that software malfunctioning causes the generation of a service message.

APPENDIX B

ADP CONTROLLED ENVIRONMENT SPECIFICATION

Security Control System

The system will provide a means to allow users to process information concurrently while providing reasonable assurance that no unauthorized release of information shall take place. The security features must be an integral part of the operating system. The contractor can assume that the physical installation will be secured to the highest level of information in the system.

The philosophy of the secure system will be such that the system will control the various shared resources. Hence the user will only be able to influence allocation decisions in a secondary way. Specifically, he can ask for a resource but not control the absolute time or address of of the resource. It is essential that any other paths which might allow the user to access information (from any device) without the access controls of the system be eliminated.

The proposed system should allow for separation of jobs (processes) in any of seven authorization levels and 16 categories to aid the user who wishes to operate at some specific classification level and category and be confident that he does not access information to which he does not have authorization.

It is recognized that to preserve security in the face of an active malicious user requires the formal certification of the correctness of the access controls. Therefore, the system shall be run in a closed controlled environment in which all users are administratively known to be benign. However, the security controls of the system must be capable of allowing users of different authorizations to process concurrently while preventing the release of information to unauthorized users.

Clearance

In this specification, clearance is defined as the eligibility of a person (or process or job) to access information of a certain classification level (or lower). For example, a person with a Secret clearance is eligible to access information with classification levels Unclassified to Secret, but may not have access to Top Secret information. When compartmented security is used, a clearance also includes the categories a person is eligible to access. In addition to the eligibility

afforded a person by his clearance, he must also have the need to know the classified information before he is given access.

Category Set

In reference to a person (or process), a category set refers to the set of compartments a person is eligible to access. A compartment in this context is an orthogonal subdivision of the classification levels. A compartment is like a formal need to know authorization to information of a certain topic without consideration of classification level.

In reference to documents, files, or other objects, a category set refers to the possible information sources used to create the object. Thus, a category set with several categories or compartments would indicate that the object should be handled with the extra caution accorded to objects which would intersect the sensitive areas of each of the categories in the set.

Specification of Security Controls

The system will provide controls which will allow users to operate concurrently while preventing the release of information to unauthorized users. The system will also prevent inadvertent violation of need to know access to data. In addition to providing the primary access controls for this environment, the system will provide programs which perform subsidiary security control functions.

Access Controls

The system will contain controls which provide for separation between users of different authorizations.

The initial classification and category set of a file (segment) will be the classification of the job (process) which created the file. In order to gain access to a file the clearance level of the job must be greater than or equal to the clearance level of the file, and the category set of the file must be a subset of the category set of the job.

Each user who can modify the access to any file is responsible for determining the need to know for all users to whom he gives access.

Sharing of information between classification levels and category sets will be controlled.

A user process will be permitted to have read access to a segment with a lower classification level, provided need to know access was specifically granted and the category set of the segment is a subset of the category set of the user process.

No user process shall have write access to any segment with a lower classification level or to any segment with a category set that does not match the category set of the user process.

No user process shall have any access to any segment with a higher classification level or to any segment with a category which is not within the category set of the user process.

A special project and a uniquely identifiable terminal will exist for the system security officer and he and only he will be allowed to perform specified security related functions.

The capability must be provided for the system hardware to check the validity of all arguments utilized in calling the operating system.

Record Keeping Features

A set of programs will be provided to aid in recording and dispensing the information created by the computer.

The system will provide a security banner on all printer output which states the level of material contained in the output determined by the classification of the process. At log in time, the classification level of the process will be printed on the terminal.

The system will provide an optional command for the printer which will allow the user to supply classification labels for page headings. It will be the user's responsibility to specify the correct page heading. The system will also provide a capability for permitting a user, at his option, to automatically place classification labels on specified terminal output.

The system will provide software capability for the process controlling the line printer to automatically prepare accountability forms for all classified output.

The system will provide the capability to insure that the system printer control will, during any period of time, handle requests of only a single classification level or a specified range of classification levels.

Auditing Capability

The system will provide an automatic capability to collect and record data regarding security related actions.

The data base management capability shall provide facilities to insure that access to data base files without appropriate access privileges shall be detected, inhibited and logged. These facilities shall be provided by applying the security and access control facilities of the operating system. The data base management capability shall not attempt to duplicate facilities provided in the operating system.

The Contractor will identify the type and forms of data to be included in the security audit trails. This data will include but will not necessarily be limited to information on access violations, rejection of illegal passwords and a record of the access granted by the system security officer.

Most of the security audit data will be recorded and listed periodically. Information describing appropriate events or actions such as repeated attempts by a user submitting illegal passwords will be printed on-line at the system console requesting the attention of the system security officer.

The audit system will have a selectable capability whereby the system security officer can subject specific projects and users to more detailed auditing than would be possible for the entire community of users. For example, monitoring all directory changes of a specific user may be desired but it would be too costly to monitor all directory changes for all users.

The audit system will be designed to interface with a user process that will perform the function of monitoring the protection mechanisms of the system. This process

will monitor the hardware and software security features, attempting to detect hardware malfunctions, before they can affect the security of the system.

The system will provide analysis programs to relate various audit data to each other by statistical methods and to summarize the results in a meaningful, concise manner.

Administrative Security Support

The system will define the administrative functions of the system security officer as differentiated from the functions assigned to the system and project administrators. These functions should permit him to support his responsibility for maintaining control of user id's, passwords, and user classification level and category set.

The system will provide a capability to permit the system security officer to generate initialize and update the data bases for user id's, passwords, and user classification levels and category set.

APPENDIX C

E-4 SECURITY KERNEL SPECIFICATION

The E-4 ADP system shall be designed and implemented with effective internal access controls which prevent unauthorized access to data that would lead to a potential or actual security compromise. The internal access controls shall provide useful tools for the development of system integrity for the E-4; i.e., a high probability that the E-4 system will correctly perform its required operational capability of properly processing data in a prompt and reliable manner.

Multilevel Security Requirement

The E-4 ADP system is is intended to operate in a multilevel mode. Specifically, the system will process information of various security classifications and will transmit multilevel information to external systems with various security clearances; viz., there will be users not cleared for all the classified data in the system.

The internal access controls in E-4 processors shall be demonstrably effective for the prevention of compromise violations of classified information. A security compromise violation is defined as one or more of the following conditions taking place:

(1) A terminal or interfaced system receives a message or data element classified to a level higher than the clearance level of the terminal or interfaced system.

(2) A terminal or interfaced system receives a message or data element having a special access category not contained in the set of special access categories authorized for the terminal or interfaced system.

Effective internal access controls shall prevent inadvertent programming errors and maliciously planted software trapdoors in the uncertified operating system and applications programs (either during initial implementation or during subsequent systems maintenance/modification) from effecting an erroneous security labeling or unauthorized accessing of information, since such occurrence could result in security compromise.

E-4 Security Model

The E-4 "security model" (viz., a precise, algorithmic statement of security functions) consists of a precise expression of the requirements and definition of

Department of Defense Regulation DOD 5200.1-R governing the classification, downgrading, declassification and safeguarding of classified information. DOD 5200.1-R provides the following definition:

Information. Knowledge which can be communicated by any means (1-311).

It also includes the following policy with respect to certain official information: "To protect against actions hostile to the United States, of both overt and covert nature, it is essential that such official information be given only limited dissemination." To implement this policy, it states that such information shall be so designated as needing protection, i.e., classified. To further aid in implementing this policy the regulation states that "The dissemination of classified information orally, in writing, or by other means, shall be limited to those persons whose official duties require knowledge or possession thereof" and more specifically no person shall be eligible for access to classified information unless a determination has been made as to his trustworthiness; i.e., he has been given a security clearance.

Elements of the E-4 Model

These simple concepts of information, people, and limiting access to information, provides the basis for representing the DOD Information Security Program in terms of a model. These concepts from the regulation, therefore, become three elements of the model; viz., people, information, and a mechanism which is interposed between the two to limit access. This access control mechanism will be called the reference monitor. (The reference monitor could represent a person; e.g., a security guard.) To complete the model, we note that the regulation also has the concepts of national security significance for information and trustworthiness for people whose official duties require knowledge of such. These attributes of people and information provide the basis for deciding whether to allow any specific person access to any specific element of information. Therefore, the model must reflect the national security significance (classification) of the information, the official positions requiring access, and the trustworthiness (clearance) of persons. This leads to the final element of the information security model -- the authorization data base. These four elements (people, information, a reference monitor, and the authorization data base) are the components of a model of the information security

program.

To achieve consistent application of policy, DOD 5200.1-R states that all official information will be assigned certain national security significance attributes. One attribute is whether the information is classified or unclassified, and if classified, the level of classification, i.e., Unclassified, Confidential, Secret, and Top Secret. In addition to its classification level, classified information may also have a second attribute -- a community of interest indicator or set of such indicators.

In addition to information, this regulation also encompasses all persons who are required to access certain classified information -- people are assigned trustworthiness attributes. The first trustworthiness attribute specifies whether a person has been evaluated and determined to be eligible by proper authority (i.e., whether a person has a security clearance) and, if cleared, the level of clearance (as determined by the type of evaluation). In addition to a clearance level, a person may be given a second attribute -- authorizations to access certain formal categories of classified information (often based on his official position).

These designated attributes form the authorization data used in the reference monitor model. The E-4 security model must represent these security attributes of people and information within the E-4 system. It does this as follows. Every piece of information within the E-4 system shall be assigned an explicit security sensitivity attribute denoted here as S. S(i) will represent the classification and categories of the ith piece of information. Information in the E-4 system consists of such things as data and program modules. It also includes representations of information such as files, buffers, terminals, and I/O devices, which may serve to present, hold, or store information in E-4.

In addition to information security attributes, every "process" (8) within the E-4 system must be assigned security access level denoted C(j). C(j) will represent the clearance level and the set of community of interest designators authorized for the jth process in E-4. The

---

(8) Roughly a process is a program in execution: a "task" or an "activity" that is scheduled for execution on some processor.

reason for assigning security attributes to processes is that each process represents a sequence of human decisions for accessing information and therefore requires the security attributes (clearance level and community of interest authorization) based on the attributes of the person(s) on whose behalf it is accessing information.

The policies in DOD 5200.1-R shall be implemented in the E-4 system in terms of rules that control the access of processes to information.

Security Policy

Access. DOD 5200.1-R, using the security attributes of information, i.e. classification level and categories, and the security attributes of people, i.e. clearance level and category access set, states the policy that before granting access to information, i.e., giving the information, to someone whose official duties require it, the recipient's clearance, i.e. clearance level and category access set, must be compared with the information's classification, i.e., classification level and categories. This policy, aimed at preventing compromise, shall be enforced within the E-4 system.

In addition to the direct access policy, there is an implicit policy throughout DOD 5200.1-R that classified information must be protected from the potential for compromise; e.g., information from a Top Secret document will not be read and then written down elsewhere in memory associated with a classification level lower than Top Secret. This policy shall also be enforced in the E-4 system.

Model Representation of Compromise Prevention

The policy concerning granting access in an E-4 processor can be stated using the security attributes of information and processes. The jth process is allowed to see (read) the ith element of information only if

$$C(j) \geq S(i)$$

where $C(j) \geq S(i)$ means the clearance level of the process is greater than or equal to the classification level of the information, and the process is authorized access to the set of categories which are assigned to the information; i.e., the clearance of the process is greater than or equal to the classification of the information. For example, a process j can read an element

of information i (e.g., message, terminal, etc.) only if its clearance is equal to or greater than the classification of the information.

The policy further says that to prevent the potential for compromise the jth process is allowed to write the ith information object only if

$$S(i) \geq C(j)$$

where $S(i) \geq C(j)$ means the classification level of the information object is greater than or equal to the clearance level of the process and all categories to which the process is authorized access are assigned to the information object. For example, a process can write (output) to a terminal only if the terminal has a classification equal to or greater than the clearance of the process. Information may thus be upgraded based on its being combined with other information of a higher classification level.

In summary, the model states that each information object and process in the E-4 processors must have security attributes (classification and clearance) assigned. It further states that any internal security control (compromise prevention) mechanism must preserve the following rules:

(1) Process j may read information object i only if $C(j) \geq S(i)$.

(2) Process j may write information object i only if $S(i) \geq C(j)$.

This model shall govern the design and development of the E-4 internal security controls and shall be used as the criteria for certification of the internal security controls in the E-4 against compromise.

Internal Access Controls Functions

Internal access control functions shall be provided as the sole means for processes to access the physical resources of the E-4 ADP system, subject to the security rules described above. The internal access control functions shall provide a secure multilevel processing environment even in the presence of errors and maliciously planted trapdoors in the operating system and applications software.

A unilevel system is defined to be a computer which handles only one classification of data. (9) Internal access controls are not required in the unilevel processor for protection against compromise. All data transmitted from the processor are treated as having a classification identical to clearance of the unilevel processor, despite the contents of security labels accompanying the transmitted data.

Design of an Internal Access Control Mechanism

Internal access controls for E-4 ADP system shall be designed to provide an internal processing environment in which processes are precluded from causing security compromise, and in which information objects can be isolated from all but selected processes for purposes of integrity.

Partitioning Activity Into Processes

A process within the E-4 ADP system is defined as a task (logically related processor activity) having a logical identity, security clearance, access capabilities implied by security attributes and community of interest, and a functional responsibility.

The following objectives shall guide the structuring of the activity of the E-4 system into logically distinct and well-defined processes:

a. Comprehensibility: By structuring the computational activity of each processor into a set of well-defined, concurrently executing processes, the operation of the system shall be made comprehensible. Providing a structure of processes is distinct from providing a set of program modules. A reason for this is that a given program module may be invoked on different occasions in various system contexts -- context includes the specific reason for invoking the program, the access capabilities of the processor while executing the program, the areas of physical memory that will be accessed by the

(9) A unilevel processor actually could receive data up to and including the clearance of the processor, but once in the processor's memory the information loses its identity for security purposes and all information must be considered classified to the single level of the processor. Therefore, all data transmitted by the processor have a classification identical with the clearance of the processor.

execution of the program, and other parameters which cannot be determined from the program code itself. Whereas a program is a static list of instructions, the process which invokes a program is a dynamic entity that carries with it information reflecting the context (in particular, access rights) of a program's execution.

The E-4 hardware and software shall be designed such that multiple processes shall be capable of concurrently executing a single program module. The global activity of the E-4 system shall be comprehensible in terms of the local activity of individual system processes, rather than by requiring the analysis of all software modules, all possible interrelationships of variables, and all possible paths of execution flow between software modules.

b. Protection from Security Compromise: The design shall fulfill the E-4 operational requirements by providing processes and information objects corresponding to those of the E-4 security model. Processes and information objects shall have identities (labels) by which their security attributes are known to the internal access controls. The internal access controls shall control the attachment of information objects to the address spaces of processes, based on the rules of the security model. The internal handling of a data of a particular classification shall be delegated to a process having a corresponding clearance. The access control mechanism of the system shall establish the address space (i.e., the set of logical names referring to memory locations accessible to the processor) for that process in such a way to demonstrably preclude unauthorized exposure (viz., a security compromise) of classified information, independent of the integrity or correctness of software modules (external to the access control mechanism software) which the process may execute. For example, if a process is handling Secret data and, consequently, must read/write certain Secret classified buffer areas, then the process shall have read-only access to any required memory addresses classified at a lower level than Secret, and shall have null access (viz., be unable to read or write) to memory containing information classified at a higher level than Secret. To enforce this protection, the access control mechanism shall mediate every access request (data or instruction fetch) that the process makes to memory; in this manner the access control design shall preclude software errors and trapdoors in the operating system and applications software from causing a process to effect a security compromise.

Access Control Mechanism Characteristics

The internal access control mechanism shall mediate all requests made by processes for access to the real (i.e., physical) resources of the system to insure that access and/or downgrading of classified information leading to unauthorized disclosure cannot take place. The access control mechanism shall be implemented as a combination of hardware and software. Three criteria shall be satisfied in implementation of the access control mechanism:

(1) The access control mechanism shall be protected from modification by programs in execution external to the access control mechanism software. That is, the access control mechanism shall be tamperproof.

(2) The access control mechanism shall mediate every access made by programs in execution. It shall always be invoked. Therefore, it shall be complete and sufficient to guarantee its effectiveness in enforcing security.

(3) The access control mechanism shall be simple and precisely defined in order to allow its effective validation for correctness. (The correctness validation shall entail establishing the precise correspondence between the access control mechanism implementation and a Government approved formal specification of its function which in turn corresponds to the E-4 security model.)

The access control mechanism shall consist of three basic elements as described below -- a hardware element, a software element, and a data base element. The access control mechanism is referred to as the security kernel.

Internal Access Control Mechanism Characteristics

The security kernel shall control all processor hardware features that permit access to real resources (processor, memory, channels, etc.). Control of physical resources shall be based on hardware descriptor registers and a software data base for specifying the contents of the hardware descriptors as dictated by the E-4 security model.

Access Control Mechanism Hardware

The hardware aspect of the internal access control mechanism shall be the set of processor hardware features designed to restrict the access capabilities of the physical processor to those authorized capabilities associated with the process in execution. In order to minimize the amount of security kernel software required to provide effective access control to real resources, the following classes of processor hardware features (either hardwired or microprogrammed) shall be provided in the E-4 system.

a. Descriptor-based addressing: Each processor shall have descriptor registers which are used in dynamic address translation to map logical (i.e., virtual) addresses contained in instruction words into physical addresses which refer to real memory locations. Each available descriptor register represents an access capability to a memory segment (either data or procedure code). The authorized mode of access (i.e., read-only, read/write, execute) is specified in the descriptor. The segment number contained in the instruction word is used to select a descriptor register. A representative dynamic address translation scheme using descriptors is shown in Figure 1.

The hardware tests (for every reference to memory) to determine if:

(1) the segment is not present in main memory or does not exist in virtual address space.

(2) the offset specified in the instruction word exceeds the segment length specified in the descriptor register.

(3) the op code in the instruction word is incompatible with the access mode (read, write, or execute) specified in the descriptor.

If any of the above conditions are sensed, then a hardware fault is generated which traps to the appropriate fault handler routine (possibly a security kernel routine). Otherwise, the memory access is allowed and the instruction is executed to completion. If a sufficient number of descriptors are available then each process can be given its own set of descriptors which are active during the execution of that process. The set of active descriptors may be selected by a single base register, for example. Otherwise, descriptor registers may have to be multiplexed among processes, which implies the unloading

and loading of descriptor words (to be performed by the security kernel) to allow the blocking (defined below) and restarting of processes. The functional and real-time constraints of E-4 processors shall dictate the specific descriptor register architecture to be employed, based upon efficiency of process switching required; however, each processor shall have at least eight distinct and independent descriptor registers for each state of privilege (defined below).

b. Processor hardware support for multiple states of privilege: Each processor shall support multiple states of privilege to allow a process to acquire and relinquish access capabilities as a function of what program the process is currently executing. Examples of hardware features of this type are the master/slave mode flip-flop used to implement privileged CPU instructions, and a base-bound register pair for delimiting the region of addressable main memory while the processor is in slave mode. The features are exercised, for example, when a call is made by applications software to the operating system, accompanied by a processor state change from slave mode to master mode in order that the process can successfully execute the called supervisor program. At least three states of privilege are required for E-4 processors to provide a separation between the security kernel and the remainder of the systems software. The three states of privilege shall be used to separate applications software from operating system software, thereby providing to a process:

(1) the minimum required set of access capabilities while executing applications software.

(2) a larger set of access capabilities while executing operating system software.

(3) the maximum set of access capabilities while executing security kernel software. This maximum set of access capabilities implies direct access to the real resources of the machine in terms of read/write access capabilities to the descriptor registers themselves. In addition, any privileged instructions (e.g., halt) or instructions directly controlling hardware resources (e.g., direct I/O instructions) shall be limited (by the hardware) to execute in only the kernel state of privilege.

One approach for implementing processor states of privilege is to employ a different set of descriptor

registers for each state of privilege, per-process, as in the Multics GE-645. (10) In this GE-645 system, the software establishes several separate sets of descriptor registers (and, therefore, several address spaces) for each process. The hardware supports the efficient switching of address spaces by providing a descriptor base register (per CPU) for selecting the desired set of descriptors. In this manner a single process can efficiently switch among several states of privilege during the course of its execution. An alternative approach is to implement privileged states directly in the hardware.

c. Descriptor-based I/O: This hardware feature is desirable because it permits portions of I/O to be handled by software external to the security kernel, and therefore simplifies the security kernel. If I/O device and port-interface buffer and control registers are made addressible as main memory locations such that a descriptor register can represent an access capability to a single I/O device or port, then selected devices and ports can be attached to the address space of a process via descriptors and I/O can be delegated to processes in the same way as segments of memory. (On the other hand, if the I/O is not descriptor-based, the I/O operations shall not be permitted outside of the kernel.) Descriptor-based I/O represents a departure from traditional I/O handling, where the capability to perform any single I/O function usually implies the capability to control all I/O of the system. Implementing this feature requires that the minimum segment size be sufficiently small such that a segment can contain the buffer and control register locations of a single I/O device or port.

Access Control Mechanism Software

The security kernel routines shall provide a set of primitives (i.e., functions to be invoked) for controlling access to the real resources of the system and thereby guarantee internal security. This is accomplished through managing the descriptor registers and the data bases from which descriptors are loaded, since all access by processes to real resources are interpreted through descriptor registers. The applications and operating system software shall have no direct control over the real resources of the system, but rather shall place calls to

(10) Organick, E. I., The Multics System: An Examination of its Structure, The MIT Press, Cambridge, MA, 1972.

the security kernel requesting that real resources be made available through the initialization of descriptor registers. The security kernel determines whether or not to satisfy the request based upon clearance, classification, and need-to-know information in the security kernel data base (see below). The privileged state of the process while executing the security kernel software allows it to manipulate descriptor registers for inclusion of the requested capabilities in the address space of the process (providing that the security rules permit this inclusion). The security kernel software must be demonstrably correct (with respect to a formal model and specification of its functions) since it has sole responsibility for access control in the system.

Access Control Data Base

This data base shall contain information upon which the security kernel shall base its decisions to grant or deny each access request. Included in the data base shall be:

(1) Security clearance of all (potential) processes in the system.

(2) Classification of all addressable objects in the system. This includes all segments, files, I/O devices, ports (peculiar to each site configuration), channels, buffers, etc. (i.e., all associated groups of memory cells to which processes will request access as a unit).

(3) Access control lists for every addressable object in the system. The access control lists specify for each addressable object a list of (potential) processes allowed access (viz., need-to-know) and the specific mode(s) of access permitted (e.g., read, write, and execute). Independent of access control lists, all access requests are subject to checking of security attributes.

Security Kernel Functions

The following security kernel primitives shall be provided to support the execution of concurrent processes in E-4 processors. These primitives shall encompass all functions needed for controlling access to the system's real resources and shall provide operating system and applications software an efficient means of accessing and

managing these resources (subject to the security rules) in terms of logical resources. The entire set of access control functions required shall be provided by security kernel software. No access control functions shall exist in any other software. These primitives shall be responsible for the management of the hardware elements of the access control mechanism (i.e., descriptor registers) and for access control data bases (i.e., software tables and lists relating to need-to-know authorization, clearances, and classifications).

Create Process

A security kernel primitive shall be provided to be invoked by a process for the creation of a separate process. The set of all required processes and their respective functions, security and integrity attributes, and access requirements shall be established in the system a priori, so that the create process primitive will prepare a process having a predetermined responsibility (as communicated to the create process primitive in the logical process name passed as an argument) and which subsequently will be in a dormant state but available for activation (See wakeup below).

Delete Process

A security kernel primitive shall be provided whereby a process may invoke its own deletion (or that of a process it created) from the set of concurrent processes operating in the system.

Block and Wakeup

Security kernel primitives shall be provided to allow processes to synchronize their execution with respect to one another by manipulation of semaphore variables. Block and wakeup shall perform the traditional functions of the well-known P and V synchronizing primitives. (11) (12) The block primitive shall provide for the unbinding of the calling process from the physical processor and the

---

(11) Dijkstra, E. W., "The Structure of the "THE" Multiprogramming System", Communications of the ACM, Vol 11, No. 5, May 1968, pp 341-346.

(12) Saltzer, J. H., Traffic Control in a Multiplexed Computer System, MAC-TR-30, (Thesis), Project MAC, MIT, Cambridge, MA, July 1966.

binding to the processor of the highest priority process whose further execution is logically permissible.

Create Segment

A security kernel primitive shall be provided for the creation of logical segments (information objects) for subsequent inclusion in the address space of a process. Segments shall consist essentially of blocks of code (procedures) or blocks of data in the address spaces of processes. Data shall be read from and written to segments. Segments shall be organized a priori by logical segment names into distinct classes which are specifically tailored to the application of the E-4 system. The classes of segments shall at least include:

(1) Memory segments (i.e., message buffers, data segments, program segments, files, temporary storage areas, interprocess communication areas, etc.).

(2) I/O channel segments (i.e., addresses consisting of I/O channel buffer and control registers, and all other registers needed for controlling and/or communicating with individual peripheral I/O devices and communication links.

Delete Segment

A security kernel primitive shall be provided to remove previously created logical segments from the set of logical segments available for inclusion in the address space of a process.

Give Access

A security kernel primitive shall be provided to give need-to-know (with specified access modes; i.e., read-only, read/write, execute) for segments to processes. Explicit controls shall be formulated for specifying where the responsibility for granting need-to-know resides; i.e., which process, sets of processes, or special programs shared by processes may modify access control lists (by invoking the kernel primitive "give access") associated with segments.

Get Access

A security kernel primitive shall be provided to allow a process to be able to access, through a descriptor register, a segment for which the process has access

authorization in accordance with the E-4 security model (based on the identity of the process, the classification and categories associated with the segment, the clearance and categories associated with the process, need-to-know, and the entire set of segments currently accessible to the process through descriptor registers). This security kernel primitive shall perform the actual initializing of hardware descriptor registers such that subsequent references made by the process through the descriptor register will successfully access the requested segment.

Release Access

A security kernel primitive shall be provided to allow a process to remove a segment from its immediately addressable address space (i.e., by unbinding the segment from the descriptor register currently used to reference the segment).

Swap-In Segment and Swap-Out Segment

Security kernel primitives shall be provided to physically move segments between main memory and secondary memory. It is desirable that a demand-paging capability be provided to select segments for swapping as opposed to the less efficient approach of programmed look-ahead or overlaying. The policy for selecting segments for swapping shall be external to the security kernel, whereas the code which performs the actual swapping shall be internal to the security kernel. It is desirable that a "write-bit" be associated with each block of main memory, so that only blocks that have been modified since their last swap-in can be identified for the required physical moving activity of the swap-out function.

Reconfigure Segments

Security kernel primitives shall be provided to perform adaptation functions regarding the variable characteristics of I/O interfaces. The security kernel must be able to update its access control data base to reflect the current identification, clearance and classification, and physical address information (i.e., buffer and control register addresses of interfaced lines, channels, terminals, devices, etc.) of all I/O interfaces. These reconfiguration primitives shall maintain the binding between the physical interfaces and the logical names (i.e., segment name) used by the operating system and applications software to refer to and communicate with the I/O interfaces.

Non-Access Control Security Functions

The security kernel shall restrict the address spaces of all processes such that they will be incapable of causing a security compromise while executing operating system and applications code. There may be processes, however, which for efficiency purposes are given simultaneous read/write access to segments of various classifications while executing non-kernel programs. Since such processes require substantial Government security review, such multilevel processes shall not be included in the E-4 design without the individual, prior approval of the Government, based on a detailed justification by the contractor. These processes will be "trusted" not to downgrade classified information by reading from a higher classified segment and writing to a lower classified segment. Therefore, trusted processes must be limited to executing only programs which have been certified correct (or at least non-malicious). These processes executing certified non-kernel code shall not be allowed to manipulate the real resources of the system (that right is reserved for the security kernel alone) but they shall be able to concurrently read and write multilevel classified segments and to make decisions based upon security label information contained in headers of messages, etc. For example, a single trusted process may be responsible for handling interrupts corresponding to message buffers being filled at the line interface. This process would have the task of moving message blocks of various classifications from interface buffers to appropriate (based on security attributes) single level internal working buffers for subsequent internal handling by processes assigned to only single level processing. Greater processing efficiency can be gained (for overcoming hardware limitations) by using trusted processes in certain bottleneck area, such as interrupt handling, since less interaction is required by the security kernel to unload and load descriptor registers (as could be required, with certain hardware, if several single level processes were delegated to perform the same task.)

Certification of Internal Access Control Software

Certification of security kernel software shall be achieved through formal correctness validation procedures applied to security kernel software. A sufficient set of the formal methods discussed below shall be applied to the security kernel validation effort.

The criteria for certification of the effectiveness of the E-4 ADP system security kernel shall consost of two parts:

(1) There shall be a thorough analysis to mathematically prove that the formal specification of the security kernel is sufficient to enforce DOD 5200.1-R as described in the E-4 security model. Thorough analysis shall be applied to validate that the formal specification of the security kernel corresponds to the E-4 security model.

(2) It shall be conclusively demonstrated that the security kernel program code implements the specification and just the specification and that no other effects are implemented. Exhaustive testing of security kernel code shall be applied to validate that the security kernel code corresponds precisely to the formal specification.

These are the sole criteria of certification and the burden of proof shall be on the contractor. The criterion for the acceptance of the kernel specification is a mathematical proof of sufficiency of the kernel to insure the effectiveness of the internal access controls. The criterion for acceptance of the kernel program code is a verification that the code implements the specification and only the specification.

The effectiveness of the kernel shall be established and certified. The kernel shall be effective regardless of attempts, either accidental or malicious, to make an unintended use of the E-4 system, e.g., attempts to send information to a place where it is not eligible to go based on significance level. The following describes an approach for certification of the E-4 security kernel.

The kernel design for an E-4 processor can be developed by applying Dijkstra's levels of abstraction (13) to separate those parts of the kernel that implement the seucrity rules, information objects, and processes required by the model. The kernel design should provide for a potentially large segmented storage system. The kernel should implement separate sequential processes that cooperate and communicate subject to the rules of the model. (Formally, interprocess communication channels

---

(13) Dijkstra, E. W., "The Structure of The Multiprogramming System," Communication of the ACM, Volume II, Number 5, May 1968.

shall be treated as information objects and constrained by the security rules).

The E-4 kernel software design shall be simple; e.g., it shall be implemented by a small (several hundred line) structured computer program. It must corespond to the E-4 security model directly. However, it still shall be necessary to verify that the information objects and processes provided by the kernel are implemented correctly and subject to the controls specified by the model. Two approaches to providing this verification can be identified. The first involves recasting the security model in terms of a series of abstract levels related to one another by functional composition. (14) The top level is proven to meet (or represent) security requirements assuming certain properties of the next lower level. The next level is then proven to preserve those properties assuming certain properties of a still lower level. The process continues until a lowest level is reached and proven based on the functions available in the hardware. Each of the levels is described in terms of a set of operations quite close to those of a programming language. Thus the entire programming language and the correspondence of the code to the E-4 security model can be verified.

An alternative appraoch to proving the correctness of the kernel software is based on the work of Price. (15) This approach involves preparing a formal specification for each function of the kernel and identifying those assumptions on which the correct operation of each function depends. A proof is then constructed that demonstrates that all of the assumptions are preserved by all of the functions. Again, the descriptions of functions are close to a programming language and facilitate proof of verification of the code that implements the specified kernel design.

---

(14) Walter, K. G., et al, Primitive Models for Computer Security, ESD-TR-74-117, Jan 74.

(15) Price, William Robert, Implications of a Virtual Memory Mechanism for Implementing Protection in a Family of Operating Systems, PhD thesis, Carnegie-Mellon University, June 1973.

MISSION
OF THE
DIRECTORATE OF COMPUTER SYSTEMS ENGINEERING

The Directorate of Computer Systems Engineering provides ESD with technical services on matters involving computer technology to help ESD system development and acquisition offices exploit computer technology through engineering application to enhance Air Force systems and to develop guidance to minimise R&D and investment costs in the application of computer technology.

The Directorate of Computer Systems Engineering also supports AFSC to insure the transfer of computer technology and information throughout the Command, including maintaining an overview of all matters pertaining to the development, acquisition, and use of computer resources in systems in all Divisions, Centers and Laboratories and providing AFSC with a corporate memory for all problems/solutions and developing recommendations for RDT&E programs and changes in management policies to insure such problems do not reoccur.